FPGA Curved Track Fitter with Very Low Resource Usage

Jinyuan Wu, M. Wang, E. Gottschalk and Z. Shi
Fermi National Accelerator Laboratory, Batavia, IL 60510, USA
jywu168@fnal.gov (630)840-8911

Abstract:
Standard least square curved track fitting process is tailored for FPGA implementation. The coefficients in the fitting matrices are carefully chosen so that only shift and accumulation operations are used in the process. The divisions and full multiplications are eliminated. Comparison in an application example shows that the fitting errors of the low resource usage implementation are less than 4% bigger than the fitting errors of the exact least square algorithm. The implementation is suitable for low-cost, low-power applications such as high energy physics detector trigger systems.

Summary:
In high-energy physics experiment detectors, track fitting is normally considered as a software task in the higher level trigger stage or analysis stage. Although direct porting the fitting algorithm into today's large size FPGA is not impossible, the cost and power consumption quickly become concerns without careful resource usage control. In fact, many silicon and power consuming operations like multiplications and divisions in many algorithms can be eliminated or replaced by low resource usage operations such as shifts, additions and subtractions. A process deviating from the mathematically accurate one certainly produces less perfect results. However, significant reduction in FPGA logic elements and power consumption overweighs small imperfectness.

In this paper, we describe a curved track fitting functional block suitable for FPGA implementation. The fitting is based on standard least square algorithm with modifications on the matrix coefficients to eliminate divisions and full multiplications. The functional block is designed to match data fetching speed so that it can be used to process flowing data stream in trigger and DAQ systems.

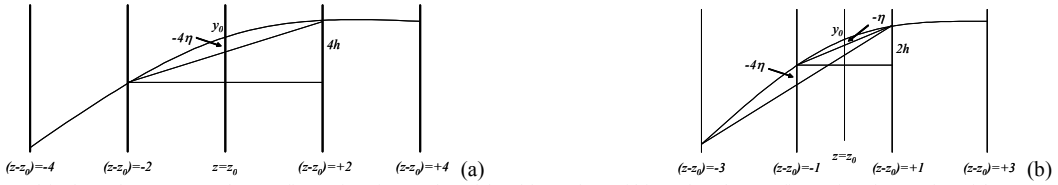

Fig. 1. Tracks in multi-plane detector: (a) the configuration for tracks with odd number of hits. (b) The configuration for tracks with even number of hits.

The FPGA track fitter was developed for the level 1 pixel trigger of the Fermilab BTeV experiment. The pixel detector consists of measurement planes as shown in Fig. 1. The hits on pixel planes are first grouped into track segments in the FPGA segment tracker using a triplet finding algorithm like Tiny Triplet Finder (TTF). Then the hits from a track are grouped together, followed by track fitting, i.e., calculating the track parameters. In the original baseline of BTeV trigger system, the track fitting is done in the trigger CPU farm partially because the hits from full detector are not switched together until reaching the CPU farm nodes. With our 2004 architecture, events are built parasitically in the early stages. Therefore, hits from full detector are available in the FPGA segment tracker, which makes it possible to perform track fitting in FPGA.

The track is projected to both the non-bend and bend views and its equations can be written approximately:

$$x = x_0 + l(z - z_0)$$

$$y = y_0 + h(z - z_0) + \eta(z - z_0)^2$$

Either a track has odd or even number of hits, a center of the track is chosen with $z=z_0$ as shown in Fig. 1. The unit in z direction is chosen so that the separation between two detector planes is 2. With a set of coordinate measurements x_i and y_i, the parameters of the tracks can be found with the following linear combinations.

$$x_0 = \frac{1}{A}\sum_i a_i x_i \quad l = \frac{1}{B}\sum_i b_i x_i$$

$$y_0 = \frac{1}{C}\sum_i c_i y_i \quad h = \frac{1}{D}\sum_i d_i y_i \quad \eta = \frac{1}{E}\sum_i e_i y_i$$

Depending on the choice of the coefficients in the linear combination, the errors of the fitting can be different and the coefficients derived from the least square fitting provide minimum errors for track reconstruction.

In general, the computations above need floating point multiplications and divisions. To simplify the computation so that it can be done in FPGA with low resource usage, the scales of the linear combinations are chosen so that:

$$xx32 = \sum_i a[i]x_i \approx 32x_0 \quad ll512 = \sum_i b[i]x_i \approx 512l$$

$$yy32 = \sum_i c[i]y_i \approx 32y_0 \quad hh512 = \sum_i d[i]y_i \approx 512h \quad eta4096 = \sum_i e[i]y_i \approx 4096\eta$$

Both the measured coordinates x_i and y_i and the coefficients in the linear combinations are integers and the results of the linear combinations: xx32, ll512, yy32, hh512 and eta4096 are also integers, representing the corresponding parameters with bit shifts. Note that divisions are not needed anymore.

To reduce computations further, the coefficients in the linear combinations are limited to the "two-bit" integers, e.g. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, which are 2^m+2^n or 2^m-2^n with integer m and n. An example of choosing e[i] for tracks with odd numbers of hits is shown in Table I. The columns of e_i in Table I represent coefficients derived from the least square fitting. The relative errors contributed by the parameter η for both algorithms are calculated. The error here is defined as transverse reconstruction RMS error after projecting the track by half-length from first or last hit of the track, with unit of the RMS error for the y_i measurements.

TABLE I
COEFFICIENTS FOR THE FPGA TRACK FITTER

	Half-length of the Track													
	16		14		12		10		8		6		4	
z-z_0	e_i	e[i]	e_i	e[i]	e_i	e[i]	e_i	e[i]	e_i	e[i]	e_i	e[i]	e_i	e[i]
-16	5.3	6												
-14	3.3	2	7.5	8										
-12	1.6	2	4.3	4	11.3	12								
-10	0.1	0	1.6	2	5.6	5	17.9	18						
-8	-1.1	0	-0.7	-2	1.0	1	7.2	7	31.0	31				
-6	-2.0	-3	-2.4	-2	-2.6	-4	-1.2	-1	7.8	8	61.0	56		
-4	-2.6	-3	-3.6	-5	-5.1	-5	-7.2	-8	-8.9	-9	0.0	12	146.3	144
-2	-3.0	-3	-4.4	-4	-6.6	-5	-10.7	-9	-18.8	-20	-36.6	-40	-73.1	-64
0	-3.2	-2	-4.6	-2	-7.2	-8	-11.9	-14	-22.2	-20	-48.8	-56	-146.3	-160
2	-3.0	-3	-4.4	-4	-6.6	-5	-10.7	-9	-18.8	-20	-36.6	-40	-73.1	-64
4	-2.6	-3	-3.6	-5	-5.1	-5	-7.2	-8	-8.9	-9	0.0	12	146.3	144
6	-2.0	-3	-2.4	-2	-2.6	-4	-1.2	-1	7.8	8	61.0	56		
8	-1.1	0	-0.7	-2	1.0	1	7.2	7	31.0	31				
10	0.1	0	1.6	2	5.6	5	17.9	18						
12	1.6	2	4.3	4	11.3	12								
14	3.3	2	7.5	8										
16	5.3	6												
Error	2.91	3.02	3.05	3.15	3.22	3.26	3.41	3.43	3.65	3.65	3.93	3.99	4.28	4.29
Ratio		1.04		1.03		1.01		1.00		1.00		1.02		1.00

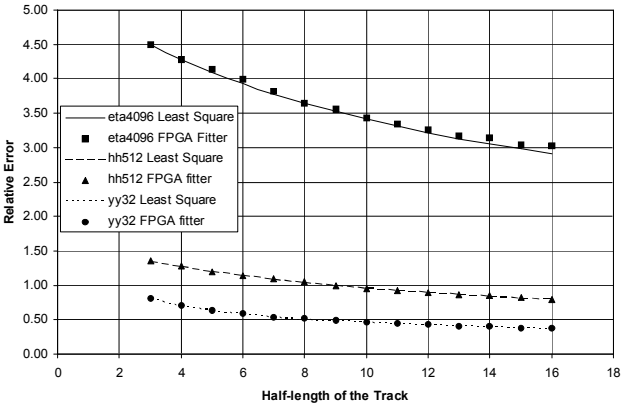

Fig. 2. Relative errors between the least square fit and the FPGA fit.

The relative errors contributed by parameters yy32, hh512 and eta4096 for both algorithms are plotted in Fig. 2. The imperfectness of coefficients for the FPGA fitting algorithm increases the track reconstruction errors but only slightly (less than %4).

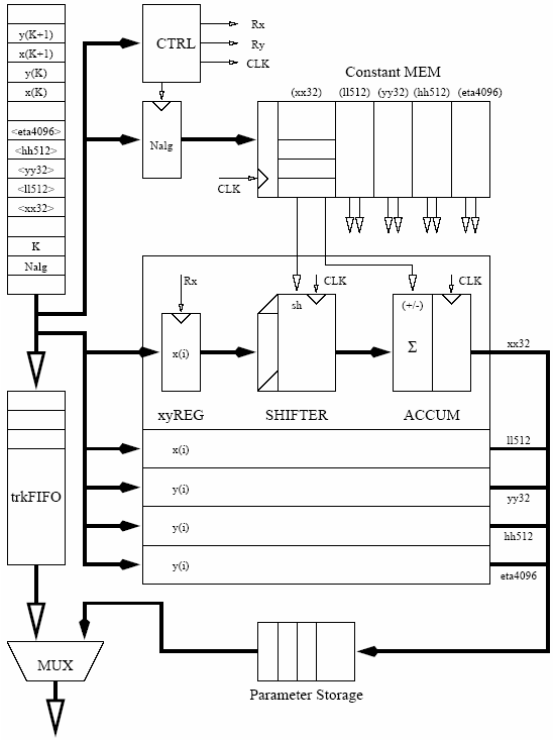

Fig. 3

The block diagram of the FPGA performing the track fitting functions is shown in Fig. 3. The hit coordinates x(i) and y(i) are clock through the fitter, one clock cycles per number.

For each parameter, an accumulator is used to calculate the linear combination. The coordinate data is shifted through a logarithmic shifter by pre-defined numbers of bits that are stored in the constant memory. The shifted version of the coordinate is added to or subtracted from the accumulator.

Each coordinate is shifted and added/subtracted twice that is equivalent to multiplying the coordinate by a two-bit integer and accumulating for the linear combination. The operation uses two clock cycles that matches the number of cycles needed to fetch a pair of coordinates x(i) and y(i).

For a fitter processing 16-bit coordinates, 630 logic elements can be accounted for in the register, shifter & accumulator blocks for the 5 parameters. The silicon resource usage of this portion is about 11% of a $30 EP1C6 device.